LEE Q63310
WIRELESS COMMUNICATION APPARATUS, WIRELESS COMMUNICATION SYSTEM ADOPTING THE SAME AND COMMUNICATION METHOD THEREOF
Filed: July 27, 2001
Darryl Mexic (202) 293-7060
1 of 9

FIG.1

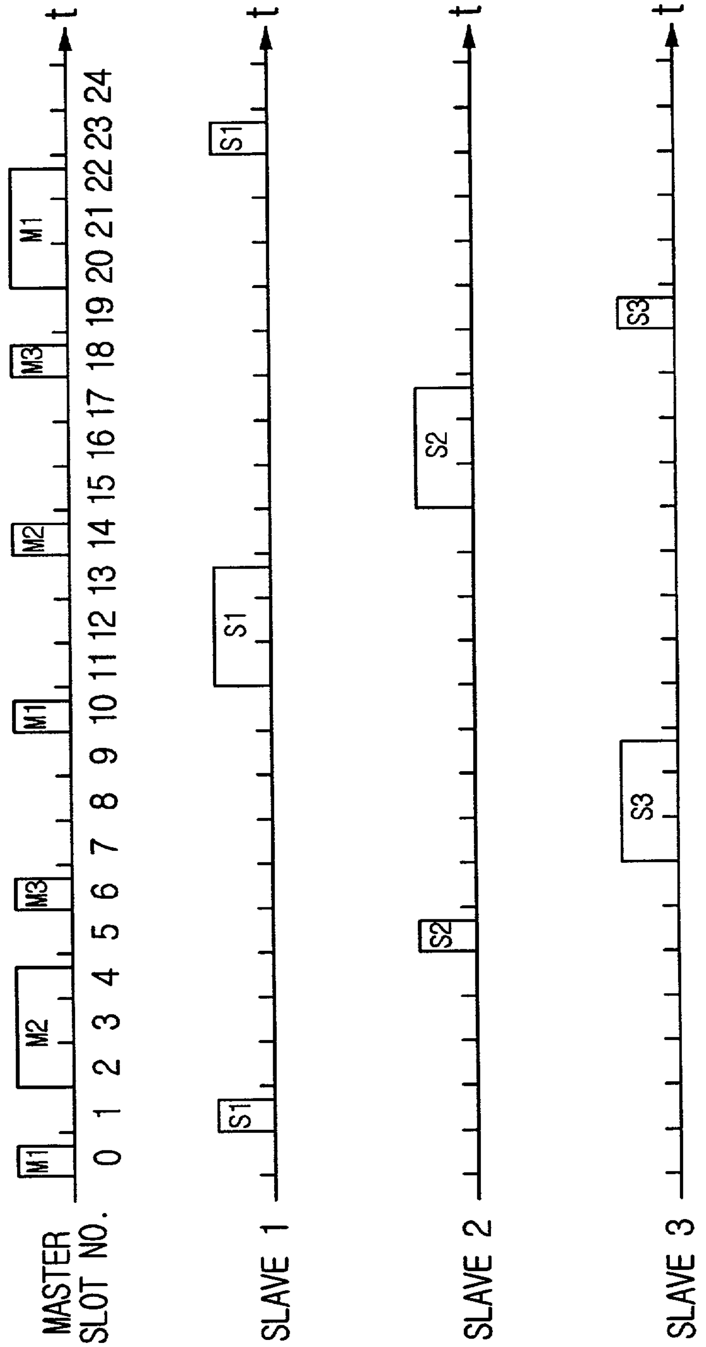


FIG.2

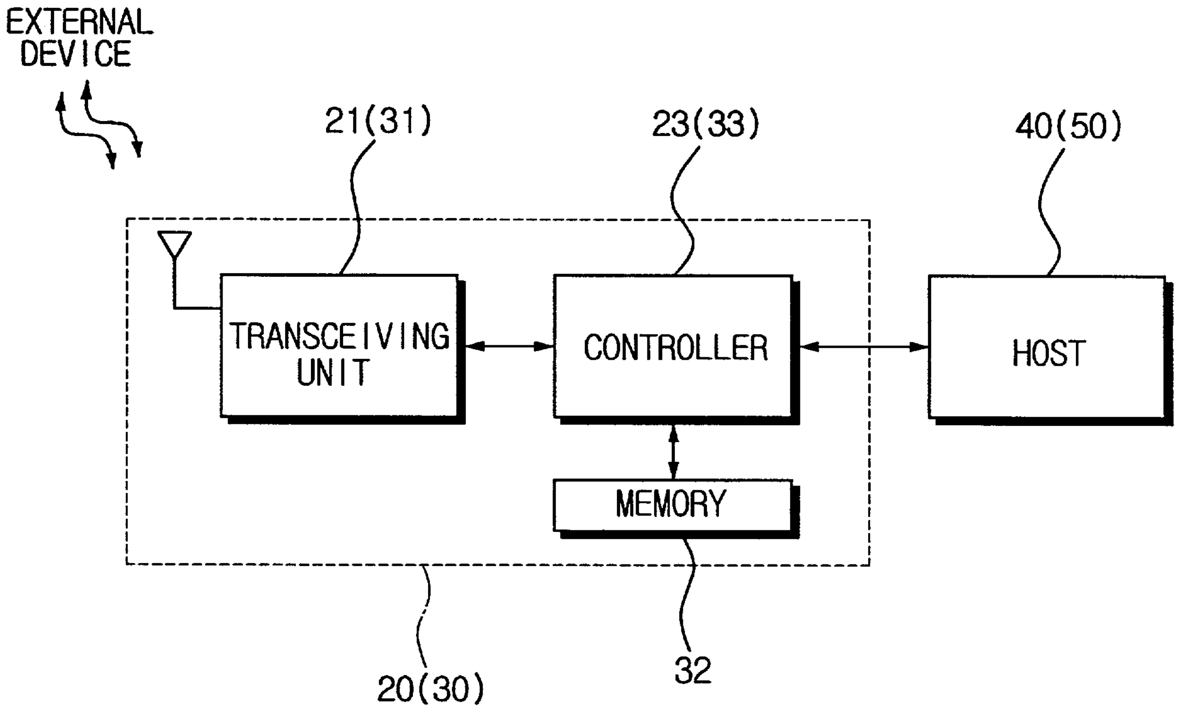

EXTERNAL DEVICE
21(31)
23(33)
40(50)
TRANSCEIVING UNIT
CONTROLLER
HOST
MEMORY
20(30)
32

FIG.3

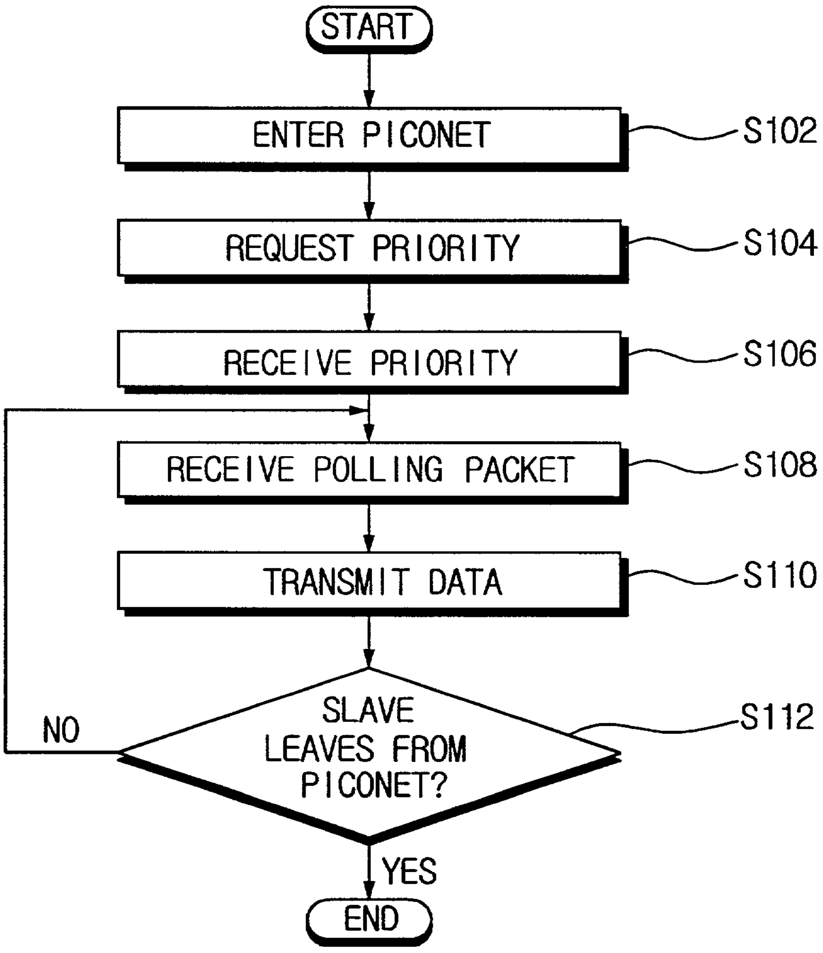

START
ENTER PICONET
S102
REQUEST PRIORITY
S104
RECEIVE PRIORITY
S106
RECEIVE POLLING PACKET
S108
TRANSMIT DATA
S110
SLAVE LEAVES FROM PICONET?
S112
NO
YES
END

FIG.4

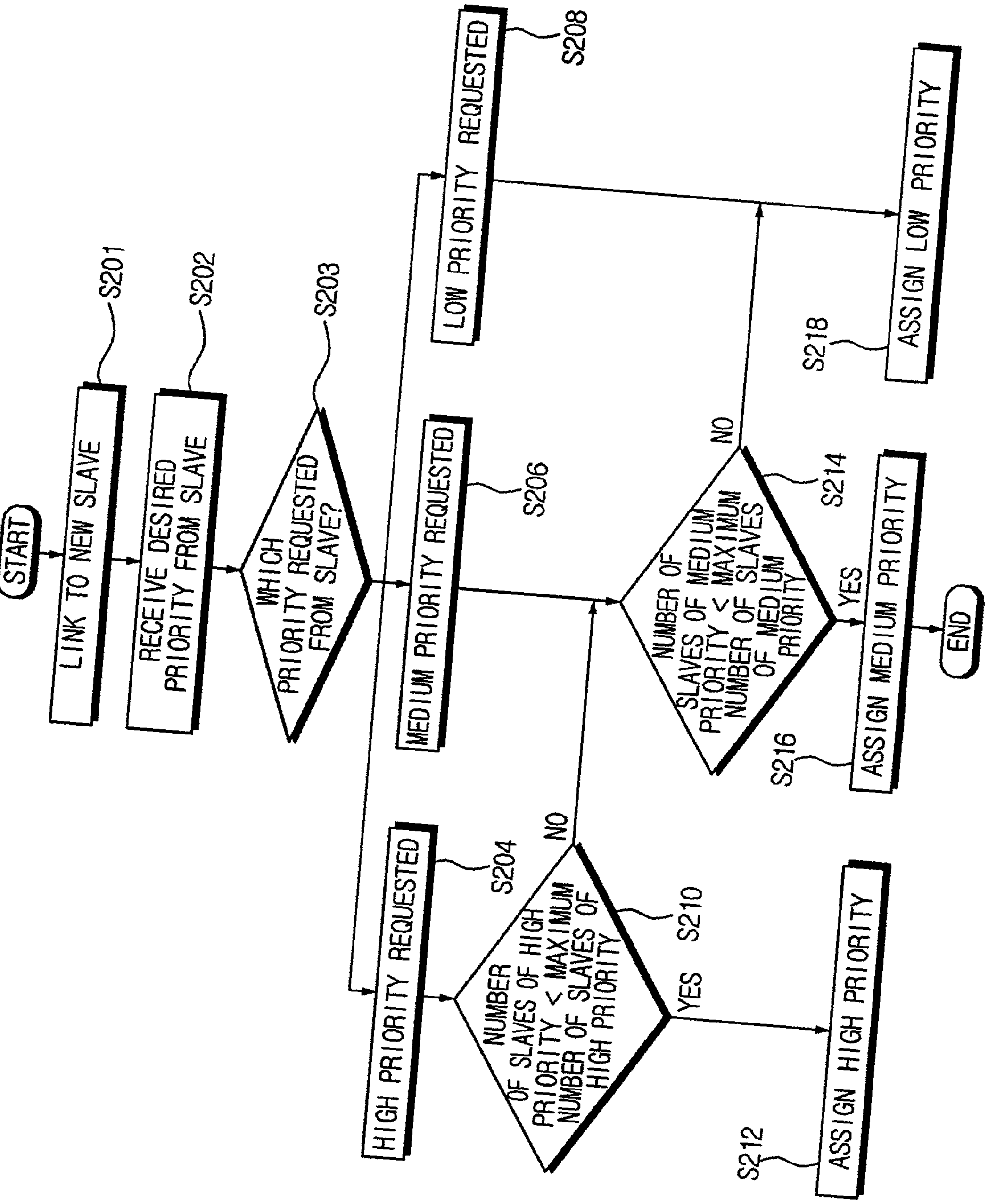
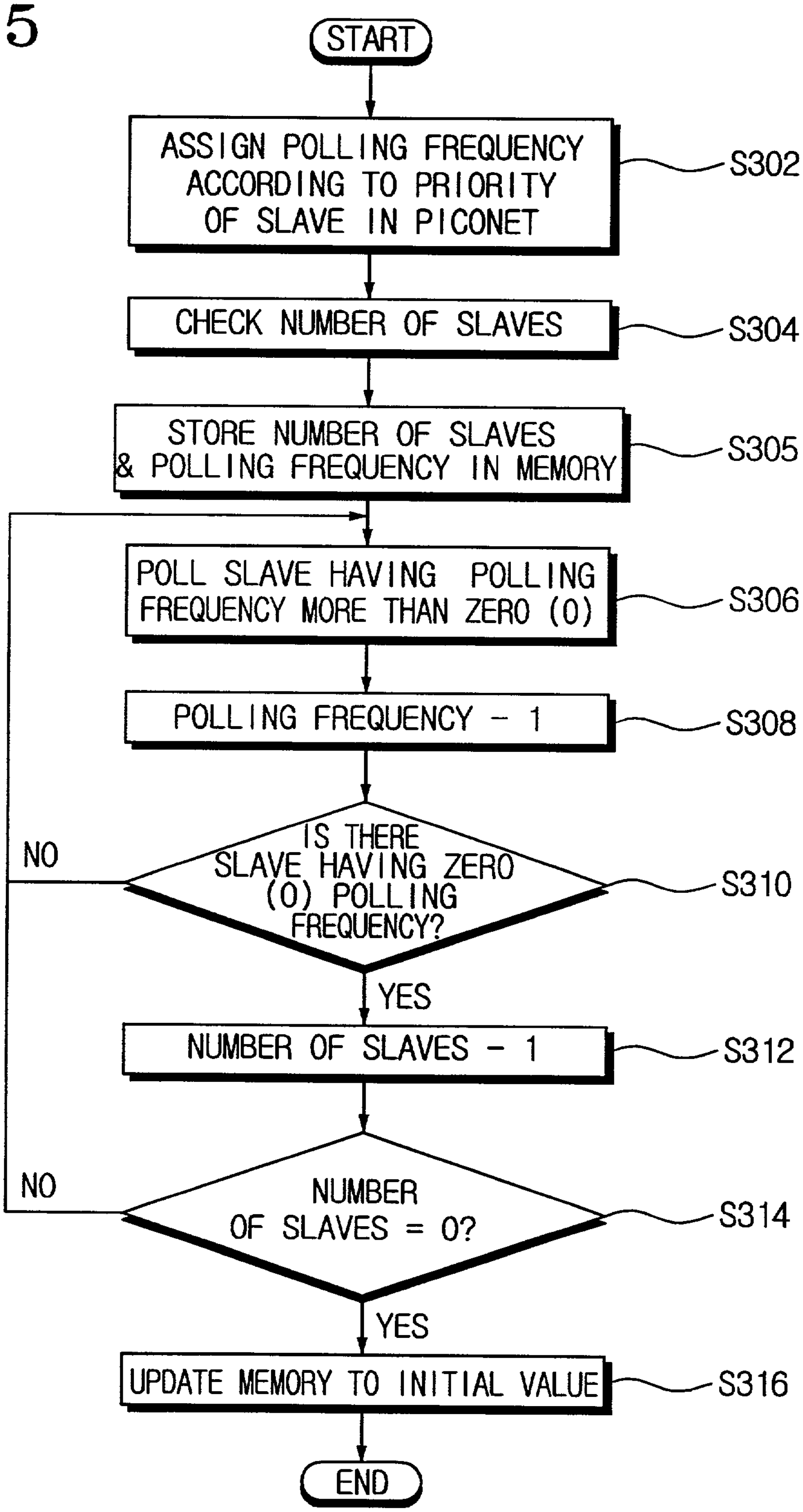

START
LINK TO NEW SLAVE
S201
RECEIVE DESIRED PRIORITY FROM SLAVE
S202
WHICH PRIORITY REQUESTED FROM SLAVE?
S203
HIGH PRIORITY REQUESTED
S204
MEDIUM PRIORITY REQUESTED
S206
LOW PRIORITY REQUESTED
S208
NUMBER OF SLAVES OF HIGH PRIORITY < MAXIMUM NUMBER OF SLAVES OF HIGH PRIORITY
S210
NO
YES
ASSIGN HIGH PRIORITY
S212
NUMBER OF SLAVES OF MEDIUM PRIORITY < MAXIMUM NUMBER OF SLAVES OF MEDIUM PRIORITY
S214
NO
YES
ASSIGN MEDIUM PRIORITY
S216
ASSIGN LOW PRIORITY
S218
END

FIG.5


START
ASSIGN POLLING FREQUENCY ACCORDING TO PRIORITY OF SLAVE IN PICONET
S302
CHECK NUMBER OF SLAVES
S304
STORE NUMBER OF SLAVES & POLLING FREQUENCY IN MEMORY
S305
POLL SLAVE HAVING POLLING FREQUENCY MORE THAN ZERO (0)
S306
POLLING FREQUENCY - 1
S308
IS THERE SLAVE HAVING ZERO (0) POLLING FREQUENCY?
S310
NO
YES
NUMBER OF SLAVES - 1
S312
NUMBER OF SLAVES = 0?
S314
NO
YES
UPDATE MEMORY TO INITIAL VALUE
S316
END

LEE Q63310
WIRELESS COMMUNICATION APPARATUS, WIRELESS COMMUNICATION SYSTEM ADOPTING THE SAME AND COMMUNICATION METHOD THEREOF
Filed: July 27, 2001
Darryl Mexic (202) 293-7060
6 of 9

FIG.6

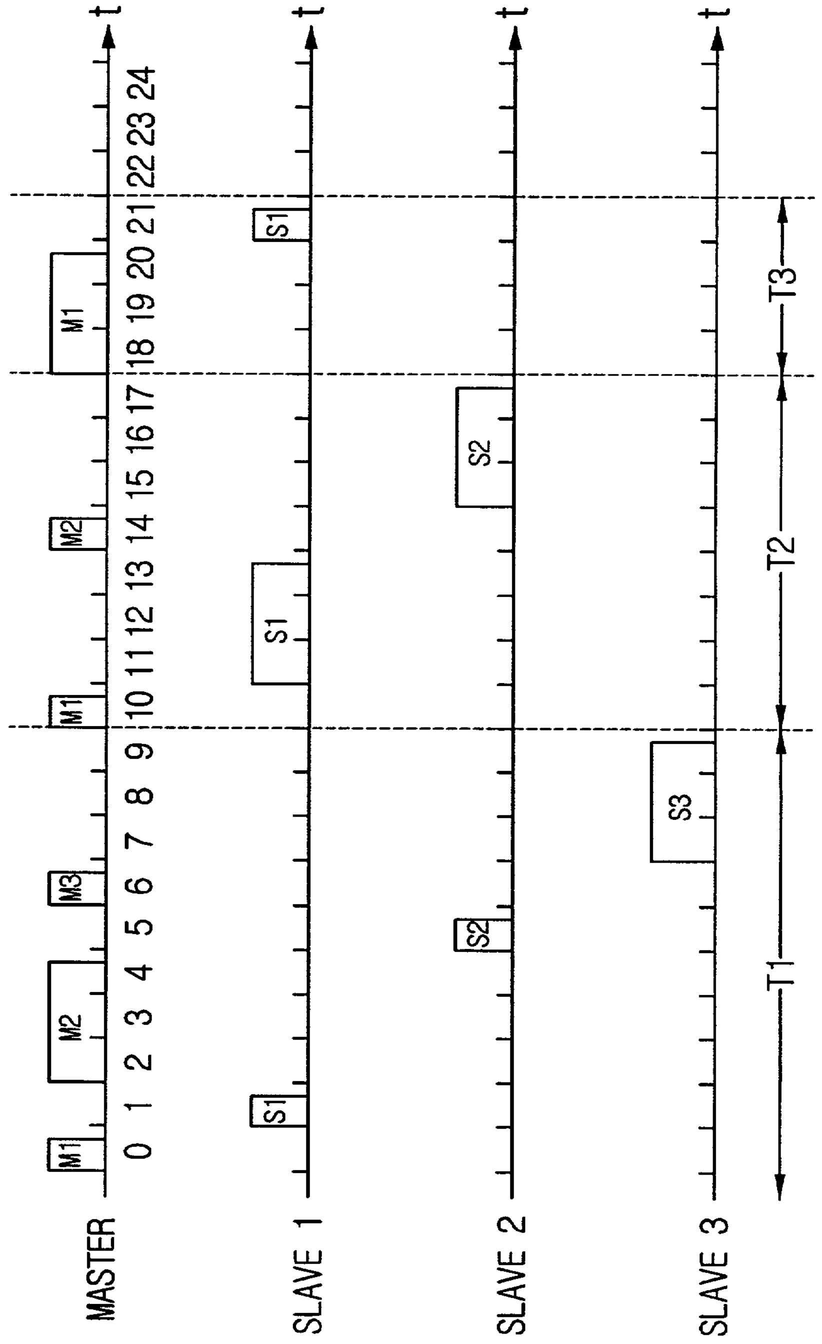


FIG.7

SLAVE	ASSIGNED PRIORITY	POLLING FREQUENCY			
		(t=0)	T1	T2	T3
SLAVE 1	HIGH	3	3→2	2→1	1→0
SLAVE 2	MEDIUM	2	2→1	1→0	•
SLAVE 3	LOW	1	1→0	•	•
COUNT (NUMBER OF SLAVES)		3	2	1	0

FIG.8A

```
MAIN:
num = 0
count = 0
while (num >= 0)
  if (new_conn ==1)                                   /*  if a new connection comes */
    num = num + 1
    DECIDE PRIORITY:
    if (new_conn_prio_request == low)                 /* if new conn's request == low priority */
      P(num) = 1                                      /* assign low priority as requested */
    else if (new_conn_prio_request == medium)         /* new conn's request == medium priority */
      if (num_of_med_prio <= 1)                       /* if number of medium priority conn <= 1 */
        P(num) = 2                                    /* assign medium priority as requested */
      else                                            /* otherwise, */
        P(num) = 1                                    /* assign low priority */
      end
    else                                              /* if new conn's request == high priority */
      if (num_of_high_prio == 0)                      /* if number of high priority conn == 0 */
        P(num) = 3                                    /* assign high priority as requested */
      else if (num_of_med_prio <= 1)                  /* if number of medium priority conn <= 1 */
        P(num) = 2                                    /* assign medium priority instead */
      else                                            /* otherwise, */
        P(num) = 1                                    /* assign low priority */
      end
    end
  end

  if (current_conn_exit == 1)                         /* if an existing connection exits */
    num = num - 1
  end
```

FIG.8B

```
  POLLING:
   if (count == 0)
     for j=1:num
       p(j) = P(j)
     end
   end

   count = num
   for j=1:num
     if (p(j)>0)
       POLL CONNECTION i
       p(j) = p(j) - 1
     else
       count = count - 1
     end
   end
end /* while */
```